

Computing Environment 100

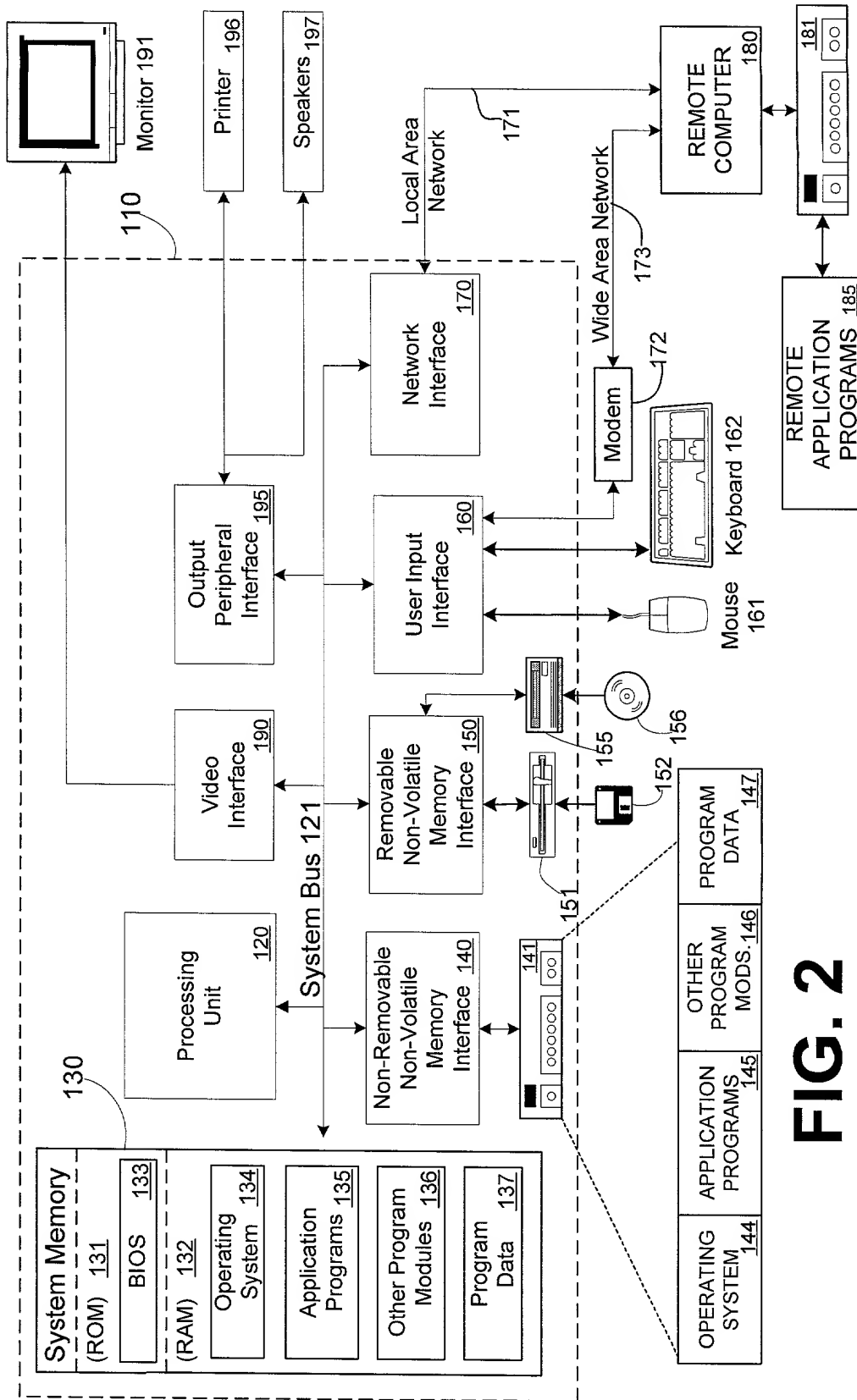


FIG. 3A

300

```
using System;
interface Interface1
{
    void F();
}
class C: Interface1
{
    public void F() {
        Console.WriteLine("C.F(), which implements Interface1.F()");
    }
}
class Test
{
    static void Main() {
        C c = new C();

        // Call through the class
        c.F();
        // Call through the interface
        Interface1 i_c = c;
        i_c.F();
    }
}
```

FIG. 3B

310

```
class C: Interface1
{
    void Interface1.F() {
        Console.WriteLine("C's implementaion of Interface1.F()");
    }
}
class Test
{
    static void Main() {
        C c = new C();

        // Can't call c.F() through the class, but
        // can still call through the interface
        Interface1 i_c = c;
        i_c.F();
    }
}
```

FD5020" E2F00660

```
interface ICloneable
{
    object Clone();
}
interface IComparable
{
    int CompareTo(object other);
}
class ListEntry: ICloneable, IComparable
{
    object ICloneable.Clone() {...}
    int IComparable.CompareTo(object
other) {...}
}
```

FIG. 4A

400

```
class Shape: ICloneable
{
    object ICloneable.Clone() {...}
    int IComparable.CompareTo(object
other) {...}
}
```

FIG. 4B

410

FIG. 4A

FIG. 4C

420

```
class Shape: ICloneable
{
    object ICloneable.Clone() {...}
}
class Ellipse: Shape
{
    object ICloneable.Clone() {...}
}
```

FIG. 4D

430

```
interface IControl
{
    void Paint();
}
interface ITextBox: IControl
{
    void SetText(string text);
}
class TextBox: ITextBox
{
    void IControl.Paint() {...}
    void ITextBox.SetText(string text) {...}
}
```

FIG. 5A

```
interface ICloneable
{
    object Clone();
}
class C: ICloneable
{
    object ICloneable.Clone() {...}
    public object Clone() {...}
}
```

500

```
interface IControl
{
    void Paint();
}
interface IForm
{
    void Paint();
}
class Page: IControl, IForm
{
    public void Paint() {...}
}
```

FIG. 5B

510

```
interface IBase
{
    int P { get; }
}
interface IDerived: IBase
{
    new int P();
}
```

FIG. 5C

520

```
class C: IDerived
{
    int IBase.P { get {...} }
    int IDerived.P() {...}
}
class C: IDerived
{
    public int P { get {...} }
    int IDerived.P() {...}
}
class C: IDerived
{
    int IBase.P { get {...} }
    public int P() {...}
}
```

FIG. 5D

530

FIG. 5C


```
interface IControl
{
    void Paint();
}
interface ITextBox: IControl
{
    void SetText(string text);
}
interface IListBox: IControl
{
    void SetItems(string[] items);
}
class ComboBox: IControl, ITextBox,
IListBox
{
    void IControl.Paint() {...}
    void ITextBox.SetText(string text) {...}
    void IListBox.SetItems(string[] items)
    {...}
}
```

FIG. 5E

540

```
interface Interface1
{
    void F();
}
class Class1
{
    public void F() {}
    public void G() {}
}
class Class2: Class1, Interface1
{
    new public void G() {}
}
```

FIG. 5F

550

```
interface IControl
{
    void Paint();
}
class Control: IControl
{
    public void Paint() {...}
}
class TextBox: Control
{
    new public void Paint() {...}
}
```

FIG. 6A

600

```
Control c = new Control();
TextBox t = new TextBox();
IControl ic = c;
IControl it = t;
c.Paint();           // invokes Control.Paint();
t.Paint();           // invokes TextBox.Paint();
ic.Paint();          // invokes Control.Paint();
it.Paint();          // invokes Control.Paint();
```

610

FIG. 6B

09900123.070501
105020.02700660

```
interface IControl
{
    void Paint();
}
class Control: IControl
{
    public virtual void Paint() {...}
}
class TextBox: Control
{
    public override void Paint() {...}
}
```

FIG. 6C

620

```
Control c = new Control();
TextBox t = new TextBox();
IControl ic = c;
IControl it = t;
c.Paint();           // invokes Control.Paint();
t.Paint();           // invokes TextBox.Paint();
ic.Paint();          // invokes Control.Paint();
it.Paint();          // invokes TextBox.Paint();
```

630

FIG. 6D

```
interface IControl
{
    void Paint();
}
class Control: IControl
{
    void IControl.Paint() { PaintControl(); }
    protected virtual void PaintControl() {...}
}
class TextBox: Control
{
    protected override void PaintControl() {...}
}
```

FIG. 6E

640

160076.1 MSFT-0573

```
interface IControl
{
    void Paint();
}
class Control: IControl
{
    void IControl.Paint() {...}
}
class MyControl: Control, IControl
{
    public void Paint() {}
}
```

700

FIG. 7A

```
interface IMethods
{
    void F();
    void G();
    void H();
    void I();
}
class Base: IMethods
{
    void IMethods.F() {}
    void IMethods.G() {}
    public void H() {}
    public void I() {}
}
class Derived: Base, IMethods
{
    public void F() {}
    void IMethods.H() {}
}
```

FIG. 7B

710

05900133.070504
FO5020.E2F00660

FIG. 7C

```
interface IBase
{
    void F();
}
interface IDerived: IBase
{
    void G();
}
class C: IDerived
{
    void IBase.F() {...}
    void IDerived.G() {...}
}
class D: C, IDerived
{
    public void F() {...}
    public void G() {...}
}
```

720

FIG. 7C is a diagram of a computer program code 720.

FIG. 8A

800

```
interface Interface1
{
    void F();
    void G();
}
```

FIG. 8B

810

```
interface Interface2
{
    void F();
    void H();
}
```

FIG. 8C

820

```
using System;
class C: Interface1, Interface2
{
    public void F() {
        Console.WriteLine("Impl of Interface1.F(),
Interface2.F()");
    }
    public void G() {
        Console.WriteLine("Impl of Interface1.G()");
    }
    public void H() {
        Console.WriteLine("Impl of Interface2.H()");
    }
}
class Test {
    static void Main() {
        C c = new C();
        c.F();
        Interface1 i1 = c;
        Interface2 i2 = c;
        i1.F();
        i2.F();
    }
}
```

0990163.070501
T05020" E3T00660

FIG. 8D

```
interface Interface1 {  
    void F();  
    void G();  
}  
interface Interface2 {  
    int F();  
    void H();  
}
```

830

FIG. 8E

```
using System;  
class C: Interface1, Interface2  
{  
    public void F() {  
        Console.WriteLine("Impl of Interface1.F()");  
    }  
    public int F() {  
        Console.WriteLine("Impl of Interface2.F()");  
        return 123;  
    }  
    public void G() {  
        Console.WriteLine("Impl of Interface1.G()");  
    }  
    public void H() {  
        Console.WriteLine("Impl of Interface2.H()");  
    }  
}
```

840

FIG. 8D

FIG. 8F

850

```
using System;
class C: Interface1, Interface2
{
    void Interface1.F() {
        Console.WriteLine("Impl of Interface1.F()");
    }
    int Interface2.F() {
        Console.WriteLine("Impl of Interface2.F()");
        return 123;
    }
    public void G() {
        Console.WriteLine("Impl of Interface1.G()");
    }
    public void H() {
        Console.WriteLine("Impl of Interface2.H()");
    }
}
class Test
{
    static void Main() {
        C c = new C();
        Interface1 i1 = c;
        Interface2 i2 = c;
        i1.F();
        int ans = i2.F();
    }
}
```


FIG. 9

900

```
class C: Interface1
{
    void Interface1.F() {
        Console.WriteLine("Impl of Interface1.F()");
    }
    void Interface1.G() {
        Console.WriteLine("Impl of Interface1.G()");
    }
    public void Dolt() {
        Console.WriteLine("C.Dolt");
    }
}
class Test
{
    static void Main() {
        C c = new C();
        c.Dolt();
        Interface1 i1 = c;
        i1.F();
    }
}
```

```
interface IEnumerator
{
    public bool MoveNext();
    public void Reset();
    public object Current { get; }
}
```

FIG. 10A

1000

```
class Item
{...}
class ItemCollection: IEnumerator
{
    // IEnumerator members
    bool MoveNext() {...}
    void Reset() {...}
    object Current { get {...} }
    // Introduced members
    void Add(Item item) {...}
    void Remove(Item item) {...}
}
```

FIG. 10B

1010

```
class ItemCollection: IEnumerator
{
    // IEnumerator members
    public bool MoveNext() {...}
    public void Reset() {...}
    object IEnumerator.Current {
        get {
            return this.Current;
        }
    }
    // Introduced members
    public void Add(Item item) {...}
    public Item Current { get {...} }
    public void Remove(Item item) {...}
}
```

FIG. 10C

1020

FIG. 10A, FIG. 10B, FIG. 10C